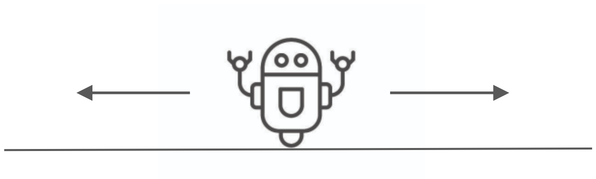
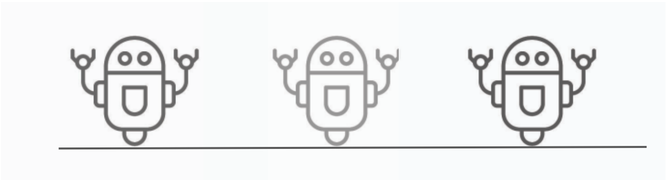
Kalman Filtering

Motivation

- Suppose we have a simple robot that can move in 1D
  - The robot starts from some initial position x0
  - We have a dynamics model of the robot but it's not precise
  - We can periodically receive ***noisy*** measurements of the robots position
- Question: Can we combine our dynamics model with the noisy measurements to get a better prediction of the robots true position?



Initial position



Position at time T?

Kalman Filtering

- The Kalman filter combines our estimation of the current state (produced by the dynamics model) with noisy measurements to produce a better estimation

Dynamics model:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

Where **F** is the state evolution model, **B** is the control input model, and **w** is the process noise (mean zero Gaussian distribution)

Kalman Filtering

Measurement model:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

Where **H** is the measurement mapping (mapping state to observation) and **v** is the measurement noise (again, mean zero Gaussian distribution)

Our goal is to make ***predictions*** with the dynamics model, and then ***update*** the predictions using measurements

Prediction Step

We first predict the current state (given the last state) using the dynamics model:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \mathbf{x}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$$

Note the process noise term is missing. The process noise is captured by the state covariance matrix given by:

$$\hat{\mathbf{P}}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k$$

Where **Q** is the covariance of the process noise (**w**). At each prediction step, we generate both the state prediction and state covariance prediction.

Update Step

Whenever a new measurement is received, we compute the difference between the measurement and the predicted measurement:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$

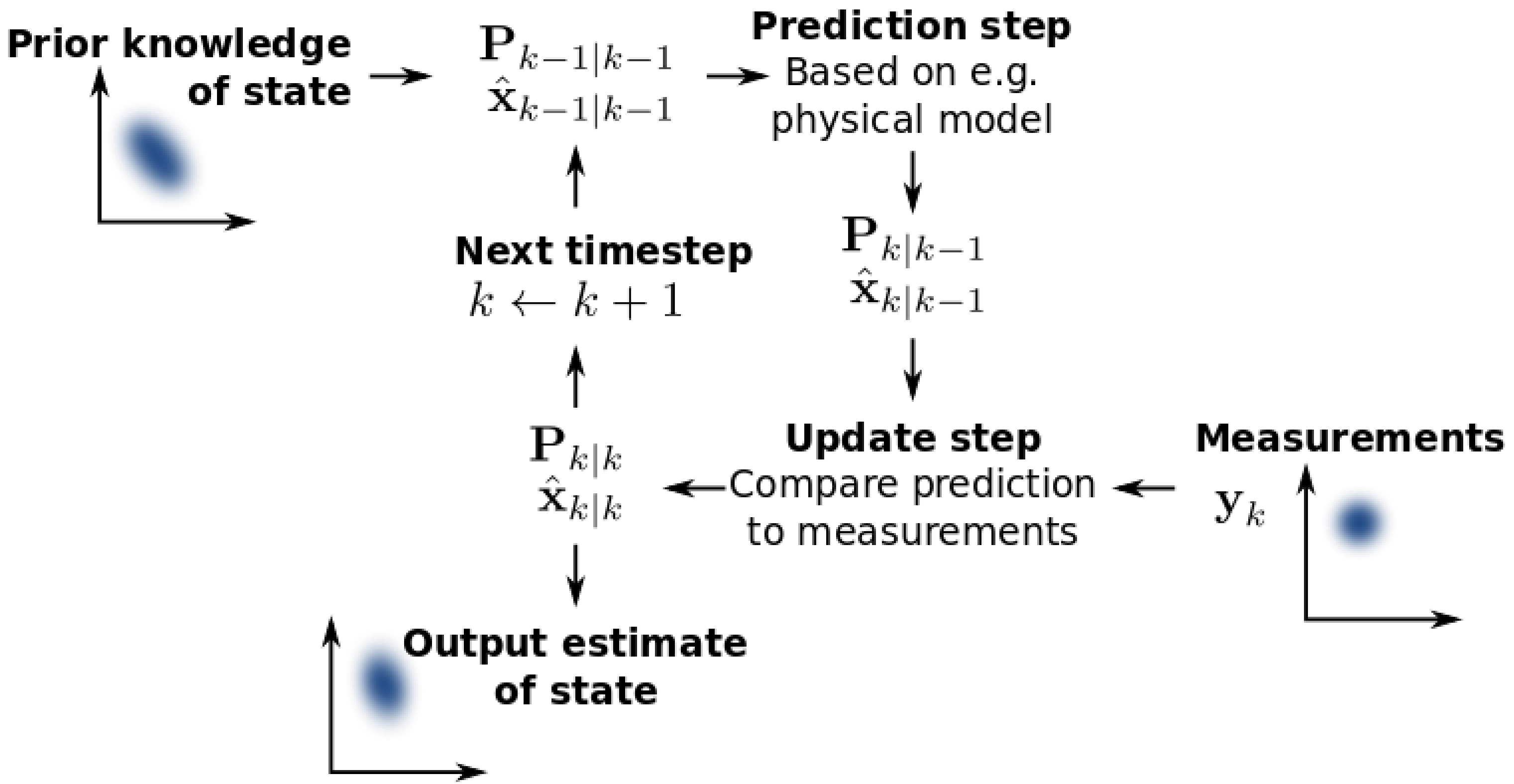
We then compute the Kalman gain (**K**) which is used to weight the update step:

$$\mathbf{K}_k = \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^\intercal (\mathbf{H}_k \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_k^\intercal + \mathbf{R}_k)^{-1}$$

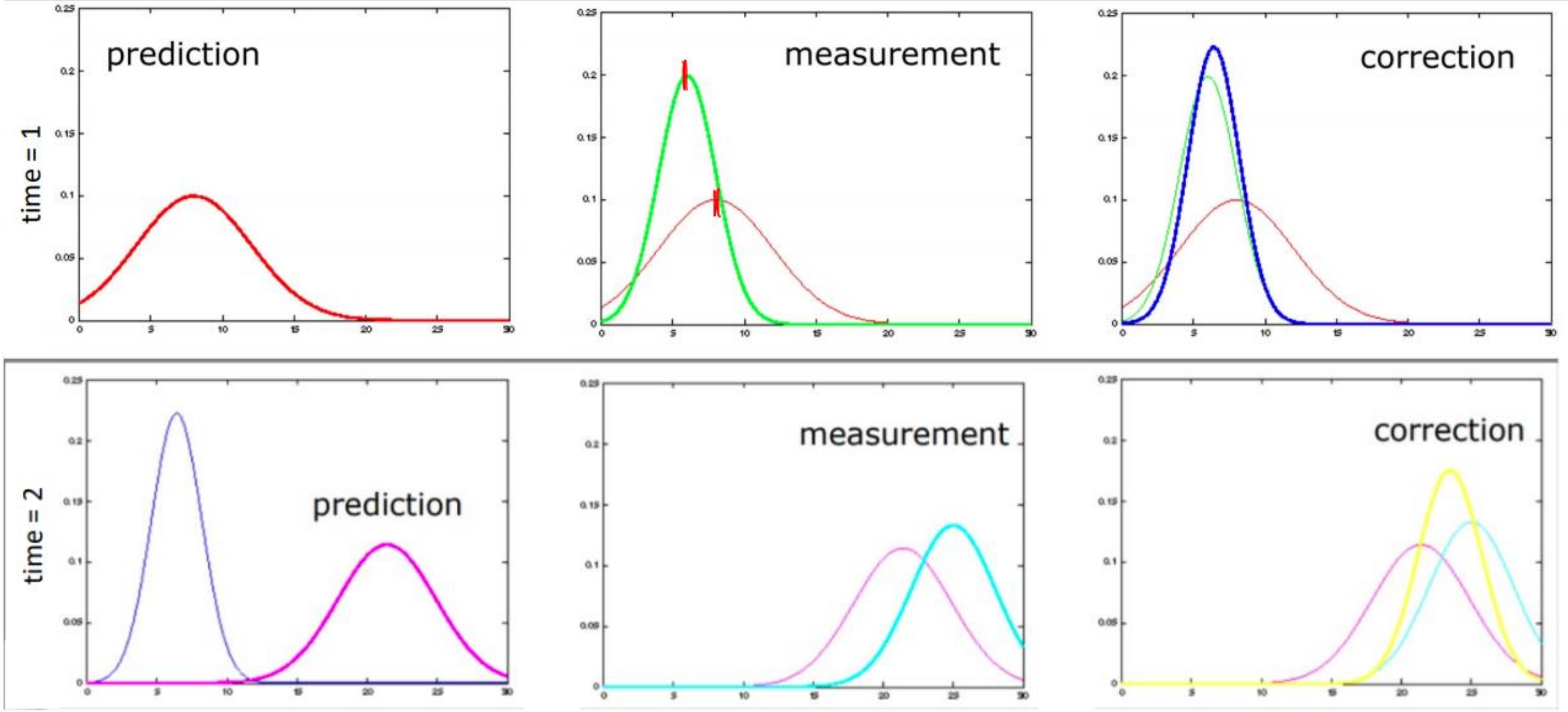
We update the predicted state and covariance with the Kalman gain:

$$\mathbf{x}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_{k|k-1}$$


Prior knowledge of state
$\mathbf{P}_{k-1|k-1}$
$\hat{\mathbf{x}}_{k-1|k-1}$
Prediction step
Based on e.g. physical model
$\mathbf{P}_{k|k-1}$
$\hat{\mathbf{x}}_{k|k-1}$
Next timestep
$k \leftarrow k+1$
Update step
Compare prediction to measurements
Measurements
$\mathbf{y}_k$
$\mathbf{P}_{k|k}$
$\hat{\mathbf{x}}_{k|k}$
Output estimate of state

1D Example


time = 1
prediction
measurement
correction
time = 2
prediction
measurement
correction

2D Example

Limitations

- Kalman filter is optimal (minimizes expected MSE) if:
  - Linear system dynamics
  - Gaussian process and measurement noise
- In practice the Kalman filter is fairly robust to non-Gaussian noise
- Modified versions of the Kalman filter exist to address non-linear systems
  - Extended Kalman filter (EKF)
  - Unscented Kalman filter